CIS 525 Parallel and distributed Software Development

General remarks on Occurrance graphs

1. T(n) and S(n) complexities for O-graphs are significant.

2. O-graph tool implemented in Standard ML and O-graph is a complex Standard ML data structure.

3. Analysis of O-graph is faster than construction.

4. O-graph can be with/without code segment with/without time.

5. O-graph as a separate program vs. integrated with CPN simulator (for instance the simulator will be able to perform an automatic simulation of an occurrence sequence found using the occurrence graph tool); also OG tool will be able to refer to the current marking of the simulator (for instance search for nodes which have an identical or similar marking).

2. Construction of O-graphs:

- To process a marking → to find the set of all enabled binding elements and the corresponding direct successor markings.
- In CPN simulator → one marking at a time.

marking = <u>a set of pointers</u> i.e. each multi-set only appears once (even so it may appear in many different markings, in many places)

marking records → page records → multi-set records

OCCURRENCE GRAPHS

1. Dynamic properties provable by Occurrance Graphs: O graph → full occurrence graph

- Reachability
- Boundedness
- Home
- Liveness
- Fairness

2. Construction of Occurrance Graph and proving dynamic properties can be fully automatic

3. Occurance Graph are defined for hierarchical CP nets

4. Sequence of presentation:

a> Example of OG using resource allocation problem
b> Definition of OG; also what is isomorphism (how to compute it?)
c> Directed paths, SCC (Strongly Connected Components)
d> Proof rules: propositions to prove/disprove CP net properties
e> Distributed DataBase
f> Dining philosophers problem
g> Construction of OG
→ how supported by computer tools
h> Analysis of OG

Remark: OG can be infinite (cycles in Resource Allocation example)

- reachability marking // text inscription of the node
- node numbers have no semantic meaning
- arc in O-graph represents the occurrence of a binding element; context of this binding element is described by the text attached to the arc

ANALYSIS OF O-GRAPHS:

- Brute force approach
- Two predefined functions:

Search Nodes
→ 6 arguments to specify details of the search
Search Arcs

Parameter 1: Search Area (specifies which part of the graph should be searched)

Parameter 2: Predicate Function (Boolean function with arguments being nodes)

Parameter 3: Search Limit (an integer; how many times the predicate function may evaluate to true before termination)

Parameter 4: Evaluation Function (a function; it maps a node into a value of some type A; this function is used only for those nodes for which predicate function evaluates to true)

Parameter 5: Start Value (a constant of some type B)

Parameter 6: Combination Function (function: A x B → B; describes how each individual result obtained by the evaluation function is combined with prior results)

Summary:

1. O-graph → node: for each reachable marking
Arc: for each occurring binding element

O-graph represents all possible occurrence sequences; i.e. both the reachable markings and occurring steps

2. Occurrence graphs are not occurrence nets

3. O-graphs have been used for arbitrary transition systems

4. Standard concepts from graph theory:

* Directed paths
* SCC → Tarjan's algorithm
* SCC-graphs
layout algorithm